UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/824,869	04/02/2001	Kevin J. McGrath	5500-64000	1001

7590 06/17/2004

Lawrence J. Merkel
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER	
GERSTL, SHANE F	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/17/2004 18

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/824,869	**Applicant(s)** MCGRATH ET AL.	
	Examiner Shane F Gerstl	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 March 2004 and 26 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,8-14,17,18 and 20-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,8-14,17,18 and 20-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02 April 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *7 and 9*.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-2, 8-14, and 17-18, and 20-37 have been examined.

Papers Received

2. Receipt is acknowledged of both information disclosure statements and amendment papers submitted, where the papers have been placed of record in the file.

3. The objections to the claims, the oath/declaration, and the 35 USC 112 rejections have been overcome by the amendment and are withdrawn.

4. Below Applicant will find rejections on the merits, for each of the claims examined, with the rejections based on Killian in the previous Office Action having been retained, with exception of those claims that have been cancelled, including clarification and with modifications necessitated by the amendment.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-2, 8-14, 17-18, 20-32 and 35 are rejected under 35 U.S.C. 102(b) as being anticipated by Killlian (5,420,992).

7. In regard to claim 1, Killian discloses a processor comprising:

a. A register configured to store a first number of bits (figure 1, element 42); In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits).

b. And an execution core (figure 1, execution unit) coupled to said register, wherein said execution core is configured to execute an instruction to produce a result, said instruction having said register as a destination, and wherein said execution core is configured to zero extend said result for update in said register responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits, and wherein said execution core is configured to preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number. Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, the register. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result into the register. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

8. In regard to claim 2, Killian discloses the processor as recited in claim 1, wherein said result comprises a number of bits specified by said operand size. The 64-bit instructions of tables 3 and 4 and discussed in columns 13-14 show that the operand size is the same as the result and thus the result is specified by the operand size.

9. In regard to claim 8, Killian discloses the processor as recited in claim 1 wherein said execution core is coupled to receive an operating mode of said processor, and wherein said execution core is configures to selectively zero extend said result further responsive to said operating mode. Column 3, lines 38-55 show that an operating mode of the processor is received. It is also shown that these modes affect the sign and zero extending configurations as shown in column 3, line 56 – column 4, line 10. Thus the zero extend functionality is responsive to the operating modes.

10. In regard to claim 9, Killian discloses the processor as recited in claim 8, wherein said operating mode includes a default operand size, and wherein said operand size corresponding to said instruction is said default operand size unless overridden by an encoding of said instruction. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size.

11. In regard to claim 10, Killian discloses the processor as recited in claim 9 wherein said execution core is configured to zero extend said result if said operand size is said default operand size. Column 3, line 67 – column 4, line 10 shows that zero extending is performed in the given operating mode. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Therefore the execution core is configured to zero extend if the operand size is the default.

12. In regard to claim 11, Killian discloses the processor as recited in claim 9 wherein said default operand size is overridden by an encoding in said instruction if said instruction includes one or more operand size prefixes. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size. A prefix is interpreted as being a set of bits. Since the instruction is a set of bits with an encoding then there is an operand size prefix included in the instruction.

13. In regard to claim 13, Killian discloses a method comprising:

a. Executing an instruction to produce a result, said instruction having a register as a destination and said register configured to store a first number of bits; Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, a register. In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits).

b. zero extending said result for update in said register responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result into the register.

c. preserving a value of at least a portion of said bits in said registers that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

14. In regard to claim 14, Killian discloses the processor as recited in claim 13, wherein said result comprises a number of bits specified by said operand size. The 64-bit instructions of tables 3 and 4 and discussed in columns 13-14 show that the operand size is the same as the result and thus the result is specified by the operand size.

15. In regard to claim 17, Killian discloses the method as recited in claim 13 wherein said zero extending and said preserving are further responsive to an operating mode of a processor performing said execution. Column 3, lines 38-55 show operating modes of a processor. It is also shown that these modes affect the sign and zero extending configurations as shown in column 3, line 56 – column 4, line 10. Thus the zero extend functionality is responsive to the operating modes. Since the intermediate extended value for the OR function will be affected by these modes in the same manner, the preserving is responsive to the modes as well.

16. In regard to claim 18, Killian discloses the method as recited in claim 17 wherein said operating mode includes a default operand size, and the method further comprises zero extending said result if said operand size is said default operand size. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Column 3, line 67 – column 4, line 10 shows that zero extending is performed in the given operating mode. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Therefore the execution core is configured to zero extend if the operand size is the default.

17. In regard to claim 20, Killian discloses an apparatus comprising:

a. a storage location corresponding to a register, said register defined to store a first number of bits (figure 1, element 42); In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits).

b. And an execution circuit (figure 1, execution unit) coupled to said storage location, wherein said execution circuit is configured to execute an instruction to produce a result, said instruction having said register as a destination, and wherein said execution core is configured to zero extend said result for update in said storage location responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits, and wherein said execution circuit is configured to preserve a value of at least a portion of said bits in said storage location that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number. Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, the register. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result into the register. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

18. In regard to claim 21, Killian discloses the apparatus as recited in claim 20 wherein said result comprises a number of bits specified by said operand size. The 64-bit instructions of tables 3 and 4 and discussed in columns 13-14 show that the operand size is the same as the result and thus the result is specified by the operand size.

19. In regard to claim 22, Killian discloses the apparatus as recited in claim 20 wherein said execution circuit is coupled to receive an operating mode, and wherein said execution circuit is configured to zero extend said result further responsive to said operating mode. Column 3, lines 38-55 show operating modes of a processor. It is also shown that these modes affect the sign and zero extending configurations as shown in column 3, line 56 – column 4, line 10. Thus the zero extend functionality is responsive to the operating modes.

20. In regard to claim 23, Killian discloses the apparatus as recited in claim 22 wherein said operating mode includes a default operand size, and wherein said operand size corresponding to said instruction is said default operand size unless overridden by an encoding of said instruction. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size.

21. In regard to claim 24, Killian discloses the apparatus as recited in claim 23 wherein said execution circuit is configured to zero extend said result if said operand size is said default operand size. The default operand size is specified

as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Column 3, line 67 – column 4, line 10 shows that zero extending is performed in the given operating mode. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Therefore the execution core is configured to zero extend if the operand size is the default.

22. In regard to claim 25, Killian discloses the apparatus as recited in claim 23 wherein said default operand size is overridden by said encoding if said instruction includes one or more operand size override prefixes. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size. A prefix is interpreted as being a set of bits. Since the instruction is a set of bits with an encoding then there is an operand size prefix included in the instruction.

23. In regard to claim 26, Killian discloses An apparatus comprising:

a. a storage location corresponding to a register, said register defined to store a first number of bits (figure 1, element 42); In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits).

b. And an execution circuit coupled to said storage location, wherein said execution circuit is configured to execute an instruction to produce a result, said instruction having said register as a destination, and wherein said execution core is configured to extend said result to said first number of bits for update in said storage location responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits, and wherein said execution circuit is configured to preserve a value of at least a portion of said bits in said storage location that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number. Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, the register. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result to 64-bits into the register. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

24. In regard to claim 27, Killian discloses the apparatus as recited in claim 26 wherein said result comprises a number of bits specified by said operand size. The 64-bit instructions of tables 3 and 4 and discussed in columns 13-14 show that the operand size is the same as the result and thus the result is specified by the operand size.

25. In regard to claim 28, Killian discloses the apparatus as recited in claim 26 wherein said execution circuit is coupled to receive an operating mode, and wherein said execution circuit is configured to selectively extend said result further responsive to said operating mode. Column 3, lines 38-55 show operating modes of a processor. It is also shown that these modes affect the sign and zero extending configurations as shown in column 3, line 56 – column 4, line 10. Thus the zero extend functionality is responsive to the operating modes.

26. In regard to claim 29, Killian discloses the apparatus as recited in claim 28 wherein said operating mode includes a default operand size, and wherein said operand size corresponding to said instruction is said default operand size unless overridden by an encoding of said instruction. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size.

27. In regard to claim 30, Killian discloses the apparatus as recited in claim 29 wherein said execution circuit is configured to zero extend said result to said first number of bits if said operand size is said default operand size. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Column 3, line 67 – column 4, line 10 shows that zero extending is performed in the given operating mode. The default operand size is specified as shown in column 3, lines 58-64, where the mode specifies the operand size via the instruction size of the program. Therefore the execution core is configured to zero extend if the operand size is the default.

28. In regard to claim 31, Killian discloses the apparatus as recited in claim 29 wherein said default operand size is overridden by said encoding if said instruction includes one or more operand size override prefixes. As shown above, the operands are of four different sizes. In order to recognize the sizes, it is inherent that there is an encoding to specify the size. A prefix is interpreted as being a set of bits. Since the instruction is a set of bits with an encoding then there is an operand size prefix included in the instruction.

29. In regard to claim 32, Killian discloses a computer system (figure 1) comprising:

a. a processor comprising a register (figure 1, element 42) configured to store a first number of bits and an execution core coupled to said register, wherein said execution core is configured to execute an instruction to produce a result, said instruction having said register as a destination, and wherein said execution core is configured to zero extend said result for update in said register responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits, and wherein said execution core is configured to preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number; In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits). Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, the register. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result to 64-bits into the register. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

b. and an input/output (I/O) device (figure 1, element 25) configured to communicate between said computer system and another computer system. Figure 1 shows that the external interface controller (I/O device) has coprocessor control and thus communicates with an external coprocessor, another computer system.

30. In regard to claim 35, Killian discloses a computer system (figure 1) comprising:

a. a processor comprising a register (figure 1, element 42) configured to store a first number of bits and an execution core coupled to said register, wherein said execution core is configured to execute an instruction to produce a result, said instruction having said register as a destination, and wherein said execution core is configured to extend said result to said first number of bits for update in said register responsive to an operand size corresponding, to said instruction specifying a second number of bits less than said first number of bits, and wherein said execution core is configured to preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number; In the embodiment introduced in column 12, lines 25-27, this register is 64-bits wide (first number of bits). Column 6, line 59 – column 7, line 7 show that the execution core (unit) executes instructions (performs operations) to produce a result that is written back to, or updates, the register. Column 12, lines 26-37 show that operands are zero extended to an operand size corresponding to an instruction. An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result to 64-bits into the register. Killian also discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

b. and an input/output (I/O) device configured to communicate between said computer system and another computer system. Figure 1 shows that the external interface controller (I/O device) has coprocessor control and thus communicates with an external coprocessor, another computer system.

Claim Rejections - 35 USC § 103

31. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

32. Claims 33 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Killian in view of IEEE (Authoritative Dictionary).

33. In regard to claim 33,

a. Killian discloses the computer system as recited in claim 32.

b. Killian does not explicitly disclose wherein the I/O device comprises a modem.

c. IEEE has disclosed on page 701 that a modem is equipment that converts data signals to appropriate form for communication between terminal equipment (computer systems) over a communication channel or medium.

d. The ability to convert data into a form appropriate for communication over a medium would have motivated one of ordinary skill in the art to modify the design of Keller to use a modem for the I/O device as taught by IEEE.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Keller to use a modem as the I/O device as taught by IEEE so that data may be transmitted between systems over a communication medium in proper form.

34. In regard to claim 36,

a. Killian discloses the computer system as recited in claim 35.

b. Killian does not explicitly disclose wherein the I/O device comprises a modem.

c. IEEE has disclosed on page 701 that a modem is equipment that converts data signals to appropriate form for communication between terminal equipment (computer systems) over a communication channel or medium.

d. The ability to convert data into a form appropriate for communication over a medium would have motivated one of ordinary skill in the art to modify the design of Keller to use a modem for the I/O device as taught by IEEE.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Keller to use a modem as the I/O device as taught by IEEE so that data may be transmitted between systems over a communication medium in proper form.

35. Claims 34 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Killian in view of Hennessy (Computer Architecture).

36. In regard to claim 34,

a. Killian discloses the computer system as recited in claim 32.

b. Killian does not explicitly disclose the system further comprising an audio device.

c. Killian has disclosed the use of an internal coprocessor (figure 1, element 22). Hennessy has taught on page 580 the use of a digital signal processor (an audio device) as a coprocessor.

d. The last paragraph of page 580 in Hennessy shows that these special-purpose computers offer higher performance and lower cost for dedicated [audio] functions such as speech recognition. The ability to manipulate audio data as has been a standard in computer processing at a high performance and low cost would have motivated one of ordinary skill in the art to modify the design of Killian to include a digital signal processor as the internal coprocessor of Killian as taught by Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Killian to include a digital signal processor as the internal coprocessor of Killian as taught by Hennessy so that audio functionality may be used in the processor, as has been standard in processors for some time, at a high performance and low cost.

37. In regard to claim 37,

a. Killian discloses the computer system as recited in claim 35.

b. Killian does not explicitly disclose the system further comprising an audio device.

c. Killian has disclosed the use of an internal coprocessor (figure 1, element 22). Hennessy has taught on page 580 the use of a digital signal processor (an audio device) as a coprocessor.

d. The last paragraph of page 580 in Hennessy shows that these special-purpose computers offer higher performance and lower cost for dedicated [audio] functions such as speech recognition. The ability to manipulate audio data as has been a standard in computer processing at a high performance and low cost would have motivated one of ordinary skill in the art to modify the design of Killian to include a digital signal processor as the internal coprocessor of Killian as taught by Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Killian to include a digital signal processor as the internal coprocessor of Killian as taught by Hennessy so that audio functionality may be used in the processor, as has been standard in processors for some time, at a high performance and low cost.

Response to Arguments

38. Applicant's arguments filed 26 March 2004 have been fully considered but they are not persuasive. Applicant's arguments regarding claim 1 state that Killian had not disclosed "zero extending said result for update in said register responsive to an operand size corresponding to said instruction specifying a second number of bits less than said first number of bits, and wherein said

execution core is configured to preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits less than said first number of bits and different from said second number."

The examiner points out that column 12, lines 26-37 of Killian show that operands are zero extended to a second operand size corresponding to an instruction (the first operand size as defined above in the rejections is 64-bits for the register size). An example of this is in the LBU (load byte unsigned) instruction of column 13, lines 10-27 and table 3A. This instruction loads a zero-extended 8-bit (second number of bits less than the first) quantity or result to 64-bits into the register. Killian discloses instances where portions of the register are preserved. An example is with the ORI (logical OR immediate) instruction of column 13, lines 36-44 and table 4A. The noted section shows that a 16-bit (third number of bits less than the first and different than the second) immediate value is combined with the register and in the case of an OR operation "OR'd" with the register contents and written to the register. The 16-bit operand is zero extended (note that this extended number is neither the result nor the operand size, but simply an intermediate value) and then logically OR'd with the contents of the register. As an example, if the hexadecimal value 0xffffffff00000000 is logically OR'd with an immediate value of 0x1234, the result will be 0xffffffff00001234 with the most significant 48 bits preserved and unchanged.

39. Arguments regarding the same above subject matter were presented with respect to claims 13, 20, 26, 32, and 35 and thus the above response is pertinent to the arguments for those claims as well.

40. Applicant also presented an argument regarding claim 1 that stated Killian does not disclose where the register is clearly the destination for the limitation that an execution core is configured to: "preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits." Applicant had asserted that the register was being used by Killian as a source register and not a destination.

The response presented above regarding other arguments shows that the register is in fact a destination register where the processor is configured to preserve a value of at least a portion of said bits in said register that are not updated by said result responsive to said operand size specifying a third number of bits since the ORI instruction writes the result to the register and as shown in the above case preserves a portion of it.

41. The examiner notes on the record that no arguments have been entered regarding the limitations not inherited by the parent claims for claims 2, 8-11, 14, and 17-18.

Conclusion

42. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

43. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

44. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The prior art made of record in the previous action remain pertinent and are cited herein.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
June 10, 2004

Eddie C

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100